Wire TAPS

THE NEWSLETTER OF THE ATARI PROGRAMMERS' SOCIETY

OCTOBER 1985 VOLUME 3 NUMBER 10

EDITORIAL

BY Peter BLake

Sometimes it is very easy to write an editorial. I know all month exactly what I want to say. The words just flow onto the screen, and before I know it, the job is done. This has not been one of those months.

It is becoming increasingly evident that my work is going to take up more and more of my time. Those who have tried to phone me know that I am seldom at home in the evenings. The few waking hours that I spend at home are usually spent working on WireTAPS. I have come to the conclusion that I no longer have the time to fulfill my duties as vicepresident of TAPS and edit WireTAPS too. After discussions with some of the members of the TAPS' executive, I have decided that it would be best if I continued to edit this newsletter. It is thus with some reluctance that I find myself having to tender my resignation as vicepresident of TAPS.

There are some regrets that I have. I had hoped that I would be on the Executive when the new constitution was adopted. But it is not to be. After more than a year in development, the new constitution is still not ready for prime time.

NOTICE OF MEETINGS

The next meeting of TAPS TORONTO will be held on Thursday October 3rd at 7:30 p.m. at the **Norseman Community School** 105 Norseman Street which is south of Bloor and east of Islington. The meeting will **NOT** be held in the resource centre. Ask for the TAPS room when you get there.

The next meeting of TAPS OSHAWA will be held on Saturday October 12th at 10:30 p.m. in the Whitby Public Library. The proposed agenda is detailed on page -3- of this issue of WireTAPS.

The next meeting of TAPS SCARBOROUGH is on Tuesday October 8th at the Cedar Ridge Crafts Centre, 255 Confederation Drive Scarborough.

WHO'S NEW

We would like to extend a special welcome to the following new members:

Michael King, Oshawa
Ernie Millard, Oshawa
Steve Glover, Whitby
John Wright, Oshawa
Ray DeNoble, Peterborough
Morris Cohen, St. Catharines
Larry Dunham, Stratford
Dan Dodds, Mississauga
Victor Brooks, Mississauga
Jim McCartney, Thornhill
Jerry Chmielewski, Toronto
Eric Cadesky, Willowdale

Welcome to all of you and your families. We look forward to seeing you at future meetings. Also a very special thanks to all the old members of TAPS who renewed this month. Your on-going support is our strength.

THEY'RE HERE!!!

A special note to all of you who have been patiently wondering where your membership cards are. The newly designed cards have arrived and anyone who joined TAPS prior to September 16 should have received theirs in this month's newsletter. If however we have neglected to include yours (in error of course) please notify us.

IN CONTROL
By Brenda Tupling

Fall is here!! With it comes the usual array of "back to school" specials and this year your local ATARI dealer has had a wide assortment of such deals for you to choose from. One of the popular choices which has attracted many members' attention includes ATARIWRITER and the ATARI 1027 printer. But what's that!!! You can't get Atariwriter to underline on your 1027? your 1025 won't print elongated letters? and your Panasonic won't utilize the AtariWriter subscript options? Welcome to the world of control codes!!!

What are control codes? They are special codes that can be typed on the keyboard but do not result in anything being printed. Instead they tell the printer to underline or skip a page; they can tell the computer to ring a bell or move the cursor around. Each printer has its own set of control codes. For those of you who have read through pages of technical jargon and are still having problems do not dispair - read on the answer you may be looking for is here.

AtariWriter & the 1027

International character sets - Before using a character from the international character set, you must type [CTRL][O]27[CTRL][O]23. This activates the international character mode. As each character is needed, you will have to use the [CTRL][O] and the decimal code for that letter (taken from the chart in the 1027 manual). To turn off the character set, type [CTRL][O]27 [CTRL][O]24.

Underlining - The 1027 will not underline when inverse video is used. Select printer number one from the main menu and insert [CTRL][O][15] where you want to begin underlining and [CTRL][O][14] where you wish to end. Your margins may have been brought in closer than the default values and if you are continuing to underline from one line to the next, the underline may start from the far left of the paper. The easiest solution for this is to adjust the margins. If this is not possible, try turning off the underlining after the last word on the line, and turn it back on before the first word on the next line.

Printing Labels - Your Atari 1027 and self adhesive address labels are not friends. The labels can easily peel off the backing and stick to the delicate inner workings of the printer. A visit to your local "authorized service dealer" may result in the entire print head mechanism being removed and replaced. $$$$$$$. Instead, consider making a label template on plain paper, and taking it to a photocopier to be reproduced using special peel-off sheets.

Printing with single sheets of letter paper - When composing and printing a document which will exceed one page, insert the [CTRL][W] page wait command at the beginning of the document under the print-formatting line. The printer will pause at the end of each page at which time you can insert a new sheet of paper. Press [RETURN] when you are ready to start the next page.

When using the print preview or print file commands, select printer option 3(820) on the print menu.

AtariWriter & Epson printers

Atari computer owners using Epson-compatible printers should respond with #3 when prompted by AtariWriter as to which printer you are using. If you wish nothing else, you will get 80 columns in normal script. However, you will get no direct response to the print commands from the AtariWriter program. To access the printer's capabilities it is necessary to send control codes by alternate means.

Firstly, it is best to edit, format and save your document. Before printing, add the required print codes at the desired place in the text. AtariWriter requires that all commands be sent in their decimal equivalent and most Epson codes must be prefaced by [ESC]. The decimal form of [ESC] is 27. Therefore each specific decimal command code should be preceded by [CTRL][O]27[CTRL][O]. The following is a table of frequently used codes.

COMPRESSED MODE:
  ON - [CTRL][O] 15
  OFF - [CTRL][O] 18
DOUBLE STRIKE MODE:
  ON - [CTRL][O] 27 [CTRL][O] 71
  OFF - [CTRL][O] 27 [CTRL][O] 72
DOUBLE WIDTH MODE:
  ON - [CTRL][O] 27 [CTRL][O] 87 [CTRL][O] 1
  OFF - [CTRL][O] 27 [CTRL][O] 87 [CTRL][O] 0
EMPHASIZED MODE:
  ON - [CTRL][O] 27 [CTRL][O] 69
  OFF - [CTRL][O] 27 [CTRL][O] 70
ITALICS MODE:
  ON - [CTRL][O] 27 [CTRL][O] 52
  OFF - [CTRL][O] 27 [CTRL][O] 53
SUBSCRIPT MODE:
  ON - [CTRL][O] 27 [CTRL][O] 84 [CTRL][O] 1
  OFF - [CTRL][O] 27 [CTRL][O] 72
SUPERSCRIPT MODE:
  ON - [CTRL][O] 27 [CTRL][O] 84 [CTRL][O] 0
  OFF - [CTRL][O] 27 [CTRL][O] 72
UNDERLINE MODE:
  ON - [CTRL][O] 27 [CTRL][O] 45
  OFF - [CTRL][O] 27 [CTRL][O] 45 [CTRL][O] 0
LINE SPACING 1/6"
  ON - [CTRL][O] 27 [CTRL][O] 50
LINE SPACING 1/8"
  ON - [CTRL][O] 27 [CTRL][O] 48
  OFF - [CTRL][O] 27 [CTRL][O] 50
LINE SPACING 7/72"
  ON - [CTRL][O] 27 [CTRL][O] 49
  OFF - [CTRL][O] 27 [CTRL][O] 50
SYSTEM RESET & NEW TOP OF FORM
  [CTRL][O] 27 [CTRL][O] 64

TAPS PETERBOROUGH

During the summer months, the TAPS Peterborough Executive have made a few changes. Vern Jackson has resigned as president and Dave Castell has assumed this position. Assisting Dave will be Grant Johnston as the newly elected Secretary/ Treasurer.

This year the Peterborough meetings will be held the second Saturday of each month at Adam Scott C.V.I., Hilliard and Langton Streets, Room 111, Peterborough. Meetings will commence at 7 p.m.

For more information call:
Dave Castell 742-8702
Grant Johston 743-0586

NEXT OSHAWA MEETING

The next meeting of TAPS Oshawa will be held Saturday October 12, in the Whitby Public Library Auditorium commencing at 10:30 a.m. The proposed agenda will be as follows:

1) opening announcements
2) 520ST demo - This time come out & see a demonstration of some of the lastest 520 software releases!!!
3) Update ATARI
4) software library demo
5) Help session

T.A.P.S. SOFTWARE LIBRARY
By Tim GLASS

Well, the summer is now over and a lot of Atari owners are now returning to their keyboards. Even though New Year's comes in January, this particular time of year always seems to me to be somewhat of a new start. I guess it stems from my memories of returning to school and beginning a 'new year' every September. Since TAPS took the summer off it now feels like we are beginning a new season, just like the schools. I thought it might be timely to review a few things about the software library, just in order to set the record straight for the months to come.

For many members, the library is the single most important aspect of their membership in the

club. It is an excellent source of economical software for the average Atari owner, and we try to provide a variety in order to appeal to everybody's interests. TAPS' software library also presents in itself a history of the progression and increasing sophistication of public-domain software for the Atari over the last few years. One need only compare some of the most recent disks with the early ones to see the difference. Not only that, but some of the present games and utilities are far more advanced than those for which we might have paid $50.00 or more in the past. It must be tough for commercial programmers who have to compete with all of the good public-domain material, most of which is free.

One is often tempted to think that most of the frontiering in this industry is over, i.e., that most of the bases have now been covered. After all, a word processor is a word processor, isn't it, and haven't we seen enough pacman-type games? Well, this may be true, but nevertheless it seems we are continually discovering new and interesting software that must be made available to all our members. With so many programmers having achieved a certain level of expertise, new and/or much improved programs are appearing everywhere, and it is the job of the software committee to attempt to procure some of the best stuff for TAPS members. We need help however, and we hope that all members are always on the lookout for programs that might be of general interest to TAPS' members. In order to encourage this process, we have a standing policy that each disk submitted to the software committee containing public-domain software for our review will entitle the member to a free copy of the latest monthly disk issue from the library. This policy does not discriminate as to the quality of these submissions--we trust that members will exercise good judgement in submitting programs that should be made available to all.

And what are the sources of this good software, you may ask. Primarily we use material which has been downloaded from BBS's such as Compuserve or the local boards. What we have been lacking in lately, however, is original material designed and/or programmed by TAPS' members themselves. I am sure that many members are working on something, or have completed programs, that perhaps they are too hesitant to display to the rest of us. We would like to encourage these members to come forward with some of their material and contribute it to the library, even if you think it's 'not good enough'. You might be surprised at how worthwhile your contributions could be. The club would really like to impress the rest of the Atari community with some of the things we can do, and the most visible way of doing this is through our production of original programs and newsletter articles. And if you have an idea for a program but are not sure whether it would fly, feel free to call me or anyone else on the executive committee to discuss it.

SOFTWARE-LIBRARY LISTINGS

In the past, members have complained about not always being provided with up-to-date listings of the software library. We had promised to provide updates via the monthly newsletter, but for various reasons this did not materialize, and I know that many members' lists are currently four to six issues behind. This year we will try again to send out the updates on a more regular basis, although it may not necessarily be monthly. Hopefully the latest update with disk numbers 24 through 29 will be included with this newsletter.

TAPS THEME LIBRARY

Earlier this year, Nick Coleman put in a lot of work in going through the whole library and picking out some of the best programs, games, and utilities that we have published over the last few years. After categorizing all of them, he came up with a group of eleven disks which will be known as the TAPS Theme Library. There should be a list of this library included with this newsletter as well.

The Theme Library will provide an alternative to those members, usually the new ones, who might wish to purchase some library selections without having to buy numerous back issues. For instance, if a member happened to be interested primarily in games and music, he could purchase what amounts to "TAPS' Greatest Hits" in those categories of the Theme Library. These disks will also be $4.00 each, the same price as our regular software library.

DOUBLE-SIDED DISKS

Several people have inquired about 'double-siding' the disks for those who are buying many at a time. Naturally we would like to help people save money where possible, and providing the Theme library is an attempt to do this. But the club officially issues only single-sided

disks, and it would be cumbersome and inconvenient for us to try to do it any other way. Of course, there is nothing to stop members from utilizing the other sides of club disks for their own purposes, but it is NOT recommended to put frequently-used material there.

THE MENU PROGRAM

At the last Toronto meeting, I was beginning to demonstrate the latest disks when a new member interrupted to inquire how the menu program works. I then realized I had been assuming that everyone knows how it works, and of course this is not necessarily so. For the benefit of those who are not very familiar with our library disks, I will just explain briefly. Each disk is 'self-booting' - just turn on your disk drive, insert the disk, and then turn on the computer. BASIC will be required for the menu program to run, so follow the appropriate procedure for booting up with BASIC. After a few seconds the disk directory will appear on the screen. The list can be scrolled through by using the SELECT key, cursor arrow keys, or joystick. Once you have made your selection, the file can be loaded and executed by pressing the START key or joystick trigger. Generally loading a file will wipe out the menu program, so returning to it afterwards must be accomplished either by pressing SYSTEM RESET or by re-booting the computer. Several other functions are incorporated into the menu program, and their use is explained on a handy 'HELP' screen which is accessible by pressing the CONTROL and 'H' keys simultaneously. I should add that this menu program is the concoction of one of our members, Roger Humphries, and has proved to be a valuable tool which many members use on all of their own disks.

THE END

Finally, I would like to thank those members from the Toronto group who volunteered to help out at the time that we were reorganizing the software library committee. For the time being it appears that we have enough manpower and equipment to get by, but perhaps other opportunities will arise for those individuals to help out the club. And we hope that everyone will consider contributing to the club in whatever way they can: whether it is software submissions, newsletter articles, demos at our meetings or just plain expertise. We can always use people's time and energy, and everyone benefits.

FRACTALS, BASIC, AND ACTION!

By Sean KEELEY

In the September 1984 issue of BYTE magazine, there was a very interesting article on fractals: mathematical entities which have recently been used to generate computer simulated landscapes, most notably by Lucasfilm. Included with the article was a short program to generate fractals, written in AppleSoft BASIC. This article will describe my experiences in implementing that program on my Atari, first in Atari BASIC, and then in Action!.

The original program was quite simple. (The listing follows this article, presumably the first Apple program to ever appear in WireTAPS!) In order to translate it to Atari BASIC, I spent a little time browsing an AppleSoft book at a local bookstore. Once I had determined how to convert the graphics and input/output statements, the rest was a direct copy (see the Atari BASIC translation, which also follows this article). The only complication occurred when I ran the program and got an error trying to take the square root of a negative number--from this I concluded that AppleSoft BASIC either carries less precision than Atari BASIC, or the square root routine treats very small negative numbers the same as zero. The fix was to insert two lines to check for the condition, and set the offending value to zero.

Running the Atari BASIC program with various inputs was quite interesting, but it was very slow, taking several minutes to produce enough points to make the figure clear. (Based on the original article, however, it did not seem to be much slower than the Apple version.) Since I also own the Action! language, I decided to implement the program in Action!, in order to get greater speed.

The process of translation to Action! was quite time-consuming. Because standard Action! does not support real numbers (numbers with decimal points, also called floating-point), it was necessary to use the real-number support routines from the Action! Programmer's Toolkit disk. Although these routines do work, they are very cumbersome to use, as can be seen from the Action! listing which also follows this article.

Imagine my surprise when, after finally finishing the Action! program, including several enhancements such as error checking and restart capability, I ran it and discovered that the

compiled code did not execute any faster than the BASIC version! Once I got over the shock of having wasted all that time, I thought about the result and realized that almost all the run time was taken up in floating-point calculations. In both BASIC and Action!, the same floating-point routines in ROM are being used, so it should have been no surprise that floating-point intensive code would execute at about the same speed in both languages.

A few months later, I purchased BASIC XL, and tried it with the fractal program. The result was as expected: it executed the program slightly faster than Atari BASIC.

As a final exercise, I decided to try and speed up the execution of the program under Atari BASIC. One of the major speed enhancements of BASIC XL is its "remembering" of GOTO and FOR loop targets; Atari BASIC must search for the target line number from the beginning of the program on each GOTO and FOR loop iteration. Because of this, one of the best ways to speed up an Atari BASIC program is to reduce the amount of this line number searching. In order to accomplish this, I moved the code which is only executed once to the end of the program, and combined many program lines to greatly reduce the total number of lines. The result was gratifying--the modified program, also listed following the article, executed essentially as fast under Atari BASIC as under BASIC XL.

So what did I learn from all this? Two things --

1. For a simple program such as this, the conversion from AppleSoft to Atari BASIC was easy.
2. Action! is not a good choice for a floating-point-intensive application, for two reasons: (a) the code is clumsy because Action! was not designed with floating-point support, and (b) there is no speed advantage when most of the time is spent in the ROM floating-point routines.

As a post-script, for anyone who keys in the optimized Atari BASIC program, the following inputs produce some interesting results. Lambda=0,1 Scale=4 produces what's called the "Dragon Curve"; Lambda=2,0 Scale=2 produces a circle; Lambda=-1,0 Scale=4 produces an "Ink Bot"; and Lambda=-2,0 Scale=3 produces a line. It should also be noted that, if too SMALL a scale value is given, the plot will be too big for the screen, and an error will occur. If this happens, try again with a LARGER value of scale.

APPLESOFT BASIC ORIGINAL

The following program is by Greg Turk, and originally appeared in the September 1984 issue of BYTE magazine. (Upper and lowercase letters have been used in all the listings to aid readability.)

```
10   Cx=140:Cy=96
20   X=0.50001:Y=0
100  Gosub 5000
110  Hgr2:Hcolor=3
120  For I=1 To 10:Gosub 4000:Next
140  Gosub 6000
160  Gosub 4000
180  Goto 140
999  End
1000 Rem SQUARE ROOT OF X,Y
1005 T=Y
1010 S=Sqr(X*X+Y*Y)
1020 Y=Sqr((-X+S)/2)
1030 X=Sqr((X+S)/2)
1040 If T<0 Then X=-X
1050 Return
2000 Rem FOUR OVER L
2010 S=Lx*Lx+Ly*Ly
2020 Lx=4*Lx/S
2030 Ly=-4*Ly/S
2040 Return
3000 Rem X,Y TIMES L
3010 Tx=X:Ty=Y
3020 X=Tx*Lx-Ty*Ly
3030 Y=Tx*Ly+Ty*Lx
3040 Return
4000 Rem FUNCTION OF X,Y
4010 Gosub 3000
4020 X=1-X
4030 Gosub 1000
4040 If Rnd(1)<0.5 Then X=-X:Y=-Y
4050 X=1-X
4060 X=X/2:Y=Y/2
4070 Return
5000 Rem GET VALUES
5010 Text:Home
5020 Input "WHAT IS LAMBDA? (X,Y) ";Lx,Ly
5030 Gosub 2000
5040 Input "WHAT IS SCALE? ";Sc
5050 Sc=2*Cx/Sc
5060 Return
6000 Rem PLOT X,Y
6010 Hplot Sc*(X-0.5)+Cx,Cy-Sc*Y
6020 Return
```

ATARI BASIC TRANSLATION

The following is a direct translation of the AppleSoft code into Atari BASIC. Other than the obvious changes required in graphics and I/O statements, the only change is the addition of statements 1015 and 1025, required to handle an apparent difference in the behaviour of the square root function with a very small negative argument.

```
10   Cx=160:Cy=96
20   X=0.50001:Y=0
100  Gosub 5000
110  Graphics 8+16:Setcolor 2,0,0:Setcolor
1,0,14:Color 1
120  For I=1 To 10:Gosub 4000:Next I
140  Gosub 6000
160  Gosub 4000
180  Goto 140
999  End
1000 Rem SQUARE ROOT OF X,Y
1005 T=Y
1010 S=Sqr(X*X+Y*Y)
1015 T2=S-X:If T2<1.0e-07 Then T2=0
1020 Y=Sqr((T2)/2)
1025 T2=S+X:If T2<1.0e-07 Then T2=0
1030 X=Sqr((T2)/2)
1040 If T<0 Then X=-X
1050 Return
2000 Rem FOUR OVER L
2010 S=Lx*Lx+Ly*Ly
2020 Lx=4*Lx/S
2030 Ly=-4*Ly/S
2040 Return
3000 Rem X,Y TIMES L
3010 Tx=X:Ty=Y
3020 X=Tx*Lx-Ty*Ly
3030 Y=Tx*Ly+Ty*Lx
3040 Return
4000 Rem FUNCTION OF X,Y
4010 Gosub 3000
4020 X=1-X
4030 Gosub 1000
4040 If Rnd(1)<0.5 Then X=-X:Y=-Y
4050 X=1-X
4060 X=X/2:Y=Y/2
4070 Return
5000 Rem GET VALUES
5010 Graphics 0
5020 ? "WHAT IS LAMBDA? (X,Y)":Input Lx,Ly
5030 Gosub 2000
5040 ? "WHAT IS SCALE?":Input Sc
5050 Sc=2*Cx/Sc
5060 Return
6000 Rem PLOT X,Y
6005 Py=Cy-Sc*Y:Px=Sc*(X-0.5)+Cx
6010 Plot Px,Py
6020 Return
```

OPTIMIZED ATARI BASIC PROGRAM

This is the program above, substantially reordered and compressed in the interests of execution speed.

```
100 Gosub 240
110 Gosub 120:Gosub 210:Goto 110
120 Rem PLOT X,Y
130 Py=Cy-Sc*Y:Px=Sc*(X-0.5)+Cx:Plot Px,Py:Return
140 Rem SQUARE ROOT OF X,Y
150 T=Y:S=Sqr(X*X+Y*Y):T2=S-X:If T2<1.0e-07
Then T2=0
160 Y=Sqr((T2)/2):T2=S+X:If T2<1.0e-07 Then
T2=0
170 X=Sqr((T2)/2):If T<0 Then X=-X
180 Return
190 Rem X,Y TIMES L
200 Tx=X:Ty=Y:X=Tx*Lx-Ty*Ly:Y=Tx*Ly+Ty*Lx:Return
210 Rem FUNCTION OF X,Y
220 Gosub 190:X=1-X:Gosub 140:If Rnd(1)<0.5
Then X=-X:Y=-Y
230 X=1-X:X=X/2:Y=Y/2:Return
240 Rem GET VALUES
250 Cx=160:Cy=96:X=0.50001:Y=0
260 Graphics 0:? "Enter Lambda (X,Y)";:Input
Lx,Ly:Gosub 320
270 ? "Enter Scale";:Input Sc
280 Sc=2*Cx/Sc
290 For I=1 To 10:Gosub 210:Next I
300 Graphics 8+16:Setcolor 2,0,0:Setcolor
1,0,14:Color 1
310 Return
320 Rem FOUR OVER L
330 S=Lx*Lx+Ly*Ly:Lx=4*Lx/S:Ly=-4*Ly/S
340 Return
```

ACTION! TRANSLATION

The following is a translation of the Atari BASIC program into the Action! programming language. As stated before, because standard Action! does not handle real numbers, the support routines from the Action! Programmer's Toolkit must be used. The need to invoke routines, even to do simple assignments, makes the program quite lengthy and hard to read. To compensate, the program is heavily commented.

Some Notes to Those Actually Reading the Code:

1. The enhancements made when the Action! program was originally written have been deleted from the listing below for comparison purposes.
2. Although my normal programming style is one statement per line, the result in this case was very long (all those procedure invocations). For publication, I have put multiple statements per line and left out some blank lines, in order to shorten the listing. Note that the colons (:) often seen separating statements in Action! listings are not used, since they serve no purpose and are ignored by the compiler.

[3.The editor has removed even more of the blank lines Sean left in in order to get everything to fit in this month's issue.]

```
; The following statement includes the Action!
; Programmer's Toolkit Real number routines.

INCLUDE "REAL.ACT"

MODULE

REAL cx,cy,x,y,tx,ty,t,t1,t2,s,lx,ly,sc,
     Re0,Rep5,Re1,Re2,Re4,ReM4

INT FUNC RealSign(REAL POINTER r)
  ; This function determines the sign of a real
number
  BYTE POINTER bptr
  bptr = r ;exponent byte of real
  IF bptr^ > 127 THEN ;negative sign
    RETURN(-1)
  ELSE
    RETURN(1)
  FI

Proc Sqrt(REAL POINTER in,out) ;Square Root
Routine
  Ln(in,out)             ;out = ln in
  RealDiv(out,Re2,out) ;out = out / 2
  Exp(out,out)           ;out = e ** out
RETURN

Proc FourOverL()
  ; s = lx * lx + ly * ly
  RealMult(lx,lx,t1) RealMult(ly,ly,t2)
  RealAdd(t1,t2,s)
  ; lx = 4 * lx / s
  RealMult(lx,Re4,lx) RealDiv(lx,s,lx)
  ; ly = -4 * ly / s
  RealMult(ly,ReM4,ly) RealDiv(ly,s,ly)
RETURN

Proc Calcxy()
  ; calculate x,y times l
  RealAssign(h,tx) RealAssign(y,ty)
  ; x = tx * lx - ty * ly
  RealMult(tx,lx,t1) RealMult(ty,ly,t2)
  RealSub(t1,t2,x)
  ; y = tx * ly + ty * lx
  RealMult(tx,ly,t1) RealMult(ty,lx,t2)
  RealAdd(t1,t2,y)
  ; x = 1 - x
  RealSub(Re1,x,x)
  ; calculate square root of x,y
  RealAssign(y,t)
  ; s = sqrt(x * x + y * y)
  RealMult(x,x,t1) RealMult(y,y,t2)
  RealAdd(t1,t2,t1)
  Sqrt(t1,s)
  ; y = sqrt((s - x) / 2)
  RealSub(s,x,t2) RealDiv(t2,Re2,t2)
  IF RealSign(t2) < 0 THEN ;s-x was very small
negative number
    RealAssign(Re0,y) ;y = 0
  ELSE
    Sqrt(t2,y) ;y = sqrt(t2)
  FI
  ; x = sqrt((s + x) / 2)
  RealAdd(s,x,t2) RealDiv(t2,Re2,t2)
  IF RealSign(t2) < 0 THEN ;s+x was very small
negative number
    RealAssign(Re0,x) ;x = 0
  ELSE
    Sqrt(t2,x) ;x = sqrt(t2)
  FI
  IF RealSign(t) < 0 THEN
    RealSub(Re0,x,x) ;x = -x
  FI
  IF Rand(0) < 128 THEN ;50/50 split
    RealSub(Re0,x,x) ;x = -x
    RealSub(Re0,y,y) ;y = -y
  FI
  RealSub(Re1,x,x) ;x = 1 - x
  RealDiv(x,Re2,x) ;x = x / 2
  RealDiv(y,Re2,y) ;y = y / 2
RETURN

Proc Plotxy()
  CARD ix
  BYTE iy
  ; ix = sc * (x - .5) + cx
  RealSub(x,Rep5,t1) RealMult(sc,t1,t1)
  RealAdd(t1,cx,t1)
  ix = RealToInt(t1)
  ; iy = cy - sc * y
```

```
  RealMult(sc,y,t1) RealSub(cy,t1,t1)
  iy = RealToInt(t1)
  Plot(ix,iy)
RETURN

Proc Initialize()
  ; Initialize FP constants + variables
  IntToReal(0,Re0) ;0 as real
  IntToReal(1,Re1) ;1 as real
  IntToReal(2,Re2) ;2 as real
  IntToReal(4,Re4) ;4 as real
  StrToReal("0.5",Rep5) ;0.5
  RealSub(Re0,Re4,ReM4) ;0-4 (-4)
  IntToReal(160,cx) ;cx=160
  IntToReal(96,cy)  ;cy=96
  RealAssign(Re0,y) ;y=0
  StrToReal("0.50001",x) ;x=.50001
  ; Get input values from user
  Graphics(0)  PrintF("%E%EEnter Lambda (X,Y)%E")
  Print("X:") InputR(lx) Print("Y:") InputR(ly)
  PrintF("%EEnter Scale:") InputR(sc)
  ; Do initial calculations on lx,ly,sc
  FourOverL()
  ; sc = 2 * cx / sc
  RealDiv(cx,sc,sc) RealMult(sc,Re2,sc)
RETURN

Proc Fractal()  ;Main Routine
  BYTE i
  Initialize()
  FOR i = 1 TO 10 DO
    Calcxy()
  OD
  Graphics(8+16)
  SetColor(2,0,0)  SetColor(1,0,15)
  color = 1
  DO
    Plotxy()
    Calcxy()
  OD
RETURN
```

WERE YOU THERE?
By Nick Coleman

If you weren't, here's (Did you know one shouldn't use conjunctions when writing? Sorry Peter. He used to be an English teacher, you know.) what happened at the TORONTO meeting on September 5th.

First of all, those who went to the meeting without reading WireTAPS, probably ended up going swimming at Norseman because the meeting was held at the Richview Public Library. It was one of the most enjoyable sessions that I've attended, possibly, in part, because I did not have the challenge of trying to look reasonably profficient while demonstrating the latest offering of the Software Library. As you all will no doubt remember, on the odd occassion... (perhaps I'm being a little conservative) ...for some inexplicable reason I did not appear to fare too well against some of my arcade advisaries; probably the different joystick... and contrary to what you may have heard rumoured, I almost never had any problem running programs on the XL machines. On this occassion, there were two, count them, two, members of the Software Library who, with great talent, successfully excuted a demonstration of the latest two Library Issues (28 & 29). From Disk #28, Tim showed his skills at BiffDrop-a levels and ladders game like Bopotron, Mission--similar to the boardgame Battleship, a double density sector copier, and DiskList-an inventory and label program written by the Software Library's Chris Thomsen. Tim also described BASIC utilities available on the disks. He nearly died from not eating enough PIE will playing DragonLord. Demonstrations from Disk #29 included an event planner that uses pull-down menus just like the 520ST, Elevator Repairman-something like Spies Demise and Ataritoons which lets you make cartoons by saving and replaying a multitude of ordered screen edit keystrokes.

We then had a break of about 45 minutes during which time, Issues #28 & #29 were sold, back orders taken, 6 new members joined and many people traded questions and answers about computing problems they had run into during the summer break. This informal exchange of information continued after the break and included a discussion on the pros and cons of various DOS's available for the ATARI and some valuble information on surge protection. Did you know your printer is more likely to be damaged by a voltage surge than than your computer? Two members indicated that it had already happened to them and another told of how their phone calls were disconnected whenever they turned their printer on!!

Doug Barton then provided a very entertaining demo of The Great North American Cross Country Car

Race by Activision ($39.99), and actually managed to beat himself at TriviaQuest by Royal Software ($39.95 U.S.).

Although the Library was available until 10:30, many of the 34 members in attendance left just after 10 o'clock. The remainder thoroughly enjoyed a demonstration of StoryFun from Disk #26, which resulted in a lot of laughter, teary eyes and sore ribs.

All in all, this was a very fun and informative meeting. Hope to see you next month, and remember only YOU can support your club and newsletter!!

FOR SALE

Great Opportunity!!

For sale: Atari 800-48K
Atari 400-64k with Bkeyboard
Gemini 10X-Epson compatible printer
Microbits-MPP 1000 modem
Atari 850 interface module
Atari 810 Disk Drive-with ARCHIVER Chip
Lots of original namebrand software
Many books and hardware manuals

Best offer over $800. If you or some one you know want to buy an ATARI computer system call Vern at 745-8026 in Peterborough. Don't wait too long!!!

FOR SALE: MPP MicroPrint Parallel Printer Interface. Least expensive way to attach a Centronics-interfaced printer to an Atari. Call Peter BLAKE at 534-3486 (Toronto).

FOR SALE: ATARI 400 complete with BASIC cartridge. Excellent condition. Asking $50. Call G. Kay (416) 723-9017 (Oshawa)

INTRODUCTION TO ASSEMBLY-LANGUAGE PROGRAMMING
Part IX - How to use Atari Floating-Point ROM

By Peter BLAKE

In Part 8 of this series, I talked about improving the graphics demo given in part 7 by the use of assembly-language subroutines. This article is about improving the graphics demo by using some of the floating-point routines that every Atari has in ROM.

In order to refresh your memory, I had some V E R Y bad code for the BASIC line "INPUT A". To save you looking it up, I shall reprint it here.

```
; :INPUT A /* This is not the bad code yet */
       LDX #0 ;IOCB #0 AGAIN
       LDA #ANSWER ;WHERE TO PUT IT
       STA ICBAL,X ;PUT IN IOCB #0
       LDA /ANSWER ;MSB OF ANSWER
       STA ICBAH,X ;PUT IN IOCB
       LDA #5 ;O.K. TO GET 5 BYTES
       STA ICBLL,X ;STOPPED BY $9B ANYWAY
       LDA #0 ;ZERO
       STA ICBLH,X ;HI-ORDER BYTE
       LDA #GETREC ;READ COMMAND
       STA ICCOM,X ;PUT IN IOCB
       JSR CIOV ;LET CIO READ ANSWER
; /* BAD CODE STARTS HERE */
       LDA ANSWER ;GET 1ST ANSWER BYTE
; In HEX 1ST byte of answer should
; be $31 (1) or $39 (9) so we shall
; AND the number in accumulator to
; get rid of four high-order bits
       AND #$0F ;ZAP OF 1ST NIBBLE
       STA GRMODE; SAVE FOR LATER
;
       LDA ANSWER+1 ;NEXT BYTE
       CMP #$9B ;ONLY ONE BYTE?
       BEQ GRSET
; Because the first digit was not the
; only digit, it should have been a
; "tens" digit so
       LDA GRMODE
       CLC ;CLEAR CARRY AS ALWAYS
       LDX #9 ;DO THIS 9 TIMES
ADDLP  ADC GRMODE
       DEX ;SUBTRACT 1 FROM X REG
       BNE ADDLP ;9 TIMES
       STA GRMODE ;SAVE 1ST DIGIT * 10
;
       LDA ANSWER+1 ;SHOULD BE 1 OR 0
       AND #$0F ;GET RID OF LEADING 3
       CLC ;INSURANCE
       ADC GRMODE
       STA GRMODE
; Now you know why people use BASIC
```

The first part of the code, up to "/* BAD CODE STARTS HERE */" is NOT bad code. In fact it is very good code as it will work on any version

of the Atari Operating System and it will not be changed. The next twenty instructions however, are a very poor way to input an integer. Hint: what is there will only work for two digits. To get around this limitation, we shall examine the Floating-Point routines, and learn a little bit about how to use them.

The floating-point chip, which occupies addresses $D800 to $DFFF contains 22 different routines. We will not look at all of them, but we will examine a few very useful ones today.

For what we are going to do today, you really do not have to know anything about floating-point mathematics. The only thing that you have to know is that a floating-point number on the Atari occupies 6 bytes. Before we begin to look at the routines, here are some very important addresses for using the Floating-Point routines:

```
FR0 = $D4 ;6-byte floating-point register 0
FR1 = $E0 ;6-byte floating-point register 1
FLPTR = $FC ;2-byte address (MSB-LSB) of a
          ;floating-point number
INBUFF = $F3 ;2-byte address (MSB-LSB) of
           ;an ATASCII string
CIX = $F2 ;1-byte offset into INBUFF
LBUFF = $0580 ;result buffer for floating-
            ;point to ATASCII conversion
```

Now we can look at some of the routines:

Routine AFP ($D800) translates an ATASCII string to an Atari Floating-point number.

Entry Conditions: INBUFF contains the address of an ATASCII numeric string.
CIX contains the offset into the string at which the number will be found. Zero if no offset.

Exit Conditions: FR0 will contain the Atari floating-point number.
CIX will point to the byte following the number
CARRY flag will be set if the number is invalid.

Routine FPI ($D9D2) converts an unsigned floating-point number to the nearest 2-byte integer.

Entry Conditions: FR0 contains a floating-point number

Exit conditions: FR0 contains a 2-byte integer in LSB-MSB format. Only the first two bytes are used.
CARRY flag will be set if the number is negative or > 65535.

Routine IFP ($D9AA) converts a two-byte unsigned integer (0 to 65535) to a floating-point number.

Entry Conditions: The 1st two bytes of FR0 contain an integer in LSB-MSB format

Exit Conditions: FR0 contains the floating-point equivalent of the integer.
If you think about it, there cannot be any errors doing this.

Routine FASC ($D8E6) converts a floating point number to its ATASCII representation.

Entry Conditions: FR0 contains a floating-point number.

Exit Conditions: INBUFF points to the first byte of the ASCII number the last byte of which has its sign bit on. No EOL ($9B) follows the string.

Now here is the code above done the right way using the floating-point routines:

```
;
; EQUATES NEEDED FOR FLOATING-POINT ROUTINES
FR0    =  $D4 ;6-BYTE FLOATING-POINT REGISTER 0
FR1    =  $E0 ;6-BYTE FLOATING-POINT REGISTER 1
FLPTR  =  $FC ;2-BYTE (LSB-MSB) POINTER TO FP NUB
INBUFF =  $F3 ;2-BYTE (LSB-MSB) POINTER TO ATASCII
;             ;INPUT BUFFER
CIX    =  $F2 ;1-BYTE OFFSET IN INBUFF
;
AFP    =  $D800 ;ATASCII TO FLOATING-POINT
;               CONVERSION ROUTINE
FPI    =  $D9D2 ;FLOATING-POINT TO INTEGER
;               CONVERSION ROUTINE
.      =  $4000 ;HI ENOUGH TO BE OUT OF DOS'S WAY
.
.  (See Summer Issue for missing code.)
.
; :INPUT A
  ... (same as before down to)
       JSR CIOV ;LET CIO READ ANSWER
; /* new code starts here */
       LDA #ANSWER ;LSB OF ANSWER
       STA INBUFF ;TELL FP WHERE TO FIND
       LDA /ANSWER ;MSB OF ANSWER
       STA INBUFF+1 ;TELL FP ROUTINES WHERE
       LDA #0 ;OFFSET TO START OF NUMBER
       STA CIX ;IN INBUFF
       JSR AFP ;CONVERT ATASCII STRING TO FLT_PNT
       BCS START ;FP ROUTINES DID NOT LIKE ANSWER
       JSR FPI ;CONVERT FP # IN REG 0 TO INTEGER
       BCS START ;FP ROUTINES DID NOT LIKE ANSWER
       LDA FR0 ; LSB OF ANSWER
       STA GRMODE ; IS ALL WE WANT
```

This code, in addition to being much shorter

than "doing it by hand" is also much more universal. It will handle integers up to 64K. Moreover, in the ten lines we have incorporated primitive error checking.

By using the routines IFP and FASC, you can convert integers in your programs to strings which can be displayed on the screen or printed on a printer. They will not work for signed numbers however. How to get around that will be dealt with much, much later.

Try playing around with the four floating-point routines presented this month. Until next time.

UPCOMING EVENTS

ComputerFest '85 is being held from the 13th to the 14th of October at the Etobicoke-Queensway Lions Club Centre. This fair is designed more for hobbyists than for business. Admission is $4.00.

The 16th annual Canadian Computer Show will be held November 18th and 19th at the International Centre on Airport Road in Malton.

The deadline for receiving articles for the Noverber issue of WireTAPS is October 18th 1985.

T H E A T A R I P R O G R A M M E R S' S O C I E T Y
P.O. Box 6287 Stn. "A" TORONTO ON M5W 1P7

THE Atari Programmers' Society (TAPS) is an independent Atari computer users' group and is not affiliated with any commercial organization. All articles in this newsletter are written by members of TAPS unless otherwise noted and no guarantees are made as to the accuracy or validity of any information contained herein. Any opinions expressed in this newsletter are those of the individual authors and do not necessarily reflect those of the THE Atari Programmers' Society. "Atari" is a trademark of Atari Inc. Any other names of products or companies are likewise trademarked and are used either with permission or as an informal method of referring to a product or company. Membership in TAPS is $25 a year.

TAPS Toronto
President: Kevin Demersen (233-4680)
Vice-Pres: Peter Blake (534-3486)
Treasurer:
Member-at-large: Tim Glass (444-1957)
Member-at-large:
BBS:

TAPS Oshawa
George Kettela (433-1678)
Paul Scudds (666-1820)
Brenda Tupling (1-668-6099)
Fabian Tapp (576-8154)
Ken Lloyd (579-1733)
668-2078 (Red Baron)

TAPS Scarborough
Graham Thorley (284-2369)
Doug Barton (757-3042)

TAPS AFFILIATED ORGANIZATIONS

Peterborough Atari Programmers (PAP)
President: Dave Castell (742-8702)
Vice-Pres:
Treasurer: Grant Johnston(743-6585)
Secretary:
Member-at-large:
BBS Sysop:
BBS:

Peninsula Atari Computer Enthusiasts (PACE)
John Daskewich (892-6014 [Fonthill])
Alan Monro (732-5027 [Welland])
Janice Munro (732-3027 [Welland])
Bart Lanni (356-7708 [Niagara Falls])
Nick Kittle (935-1500 [St.Catherines])
Brian Calcraft (357-0786) [Niagara Falls])
(357-2955 [Niagara Falls])

Permission is granted to any Atari Users' group to reprint articles contained in this newsletter on condition that credit is given to the author and to THE ATARI PROGRAMMERS' SOCIETY.

Applesoft is a registered trademark of Apple Computer Inc. Action! and BASIC XL are registered trademarks of Optimized Systems Software. Epson is a registered trademark of Epson America Inc.